

Федеральное государственное бюджетное образовательное учреждение высшего образования



**«Московский государственный технический
университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)**

ФАКУЛЬТЕТ ФУНДАМЕНТАЛЬНЫЕ НАУКИ

КАФЕДРА МАТЕМАТИКА И КОМПЬЮТЕРНЫЕ НАУКИ

Отчет

по лабораторному заданию № 8

Вариант 20

Дисциплина: Информатика

Название лабораторного задания: Динамические структуры данных. Списки.

Студент гр. ФН11-12Б

ХФ 03.12.2020

(Подпись, дата)

М.Х. Хаписов

(И.О. Фамилия)

Преподаватель Доцент кафедры ИУ-6

Т.Н. Ничушкина

(Подпись, дата)

Т.Н. Ничушкина

(И.О. Фамилия)

3.12.2020

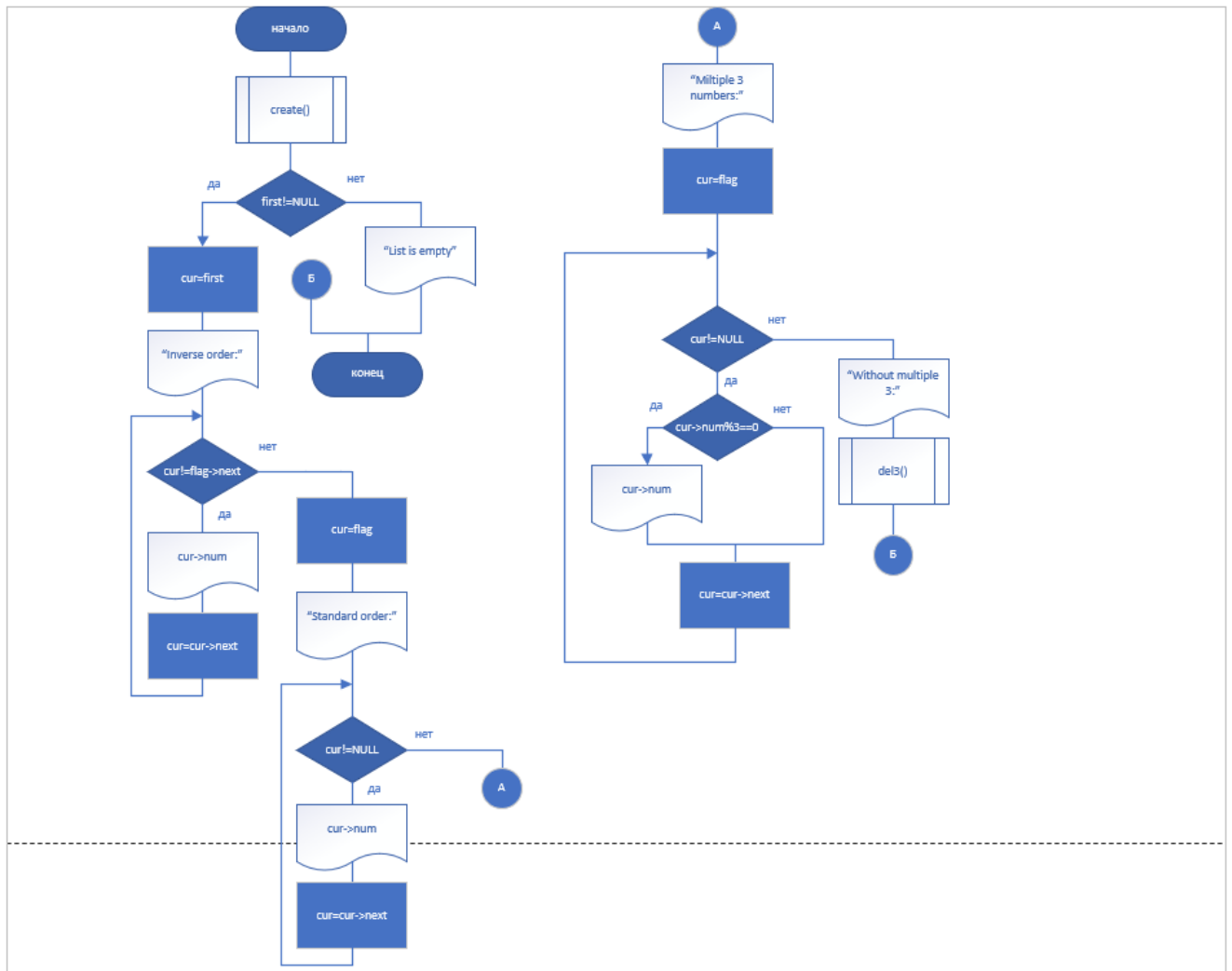
Москва, 2019

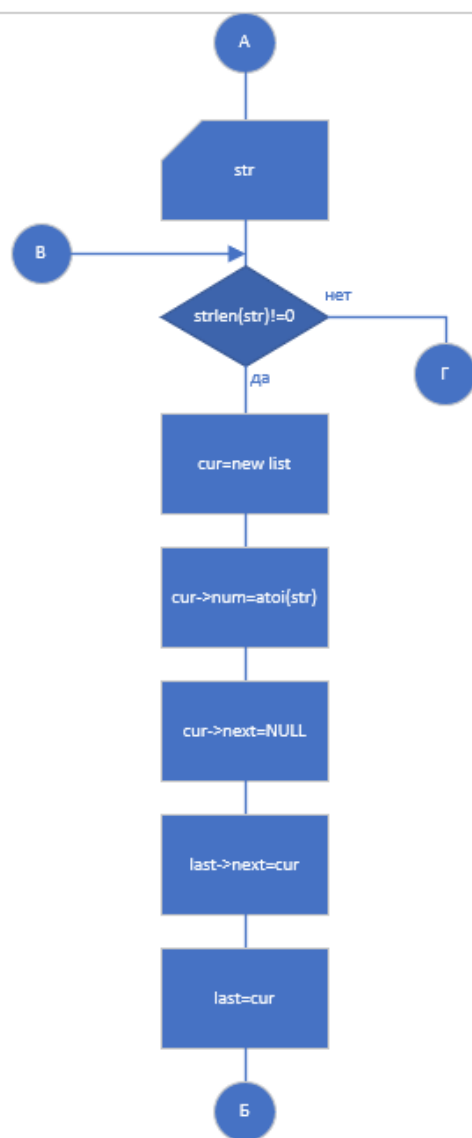
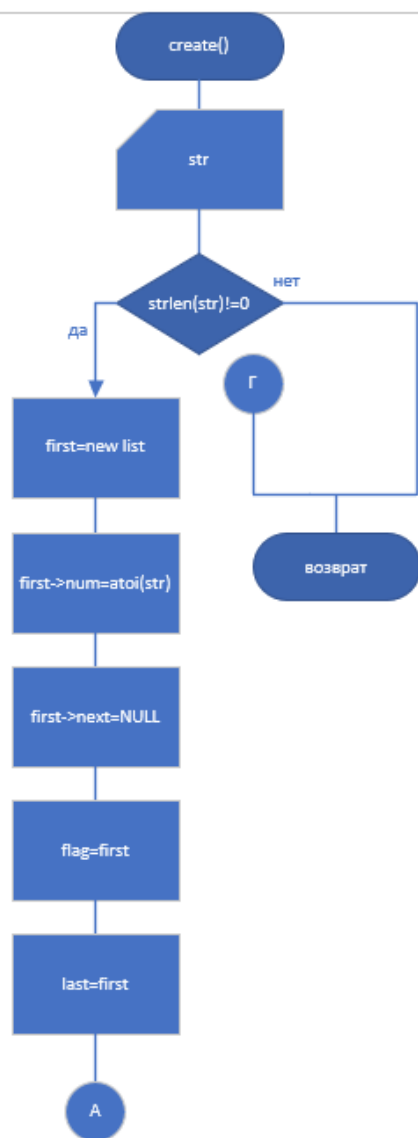
Динамические структуры данных. Списки.

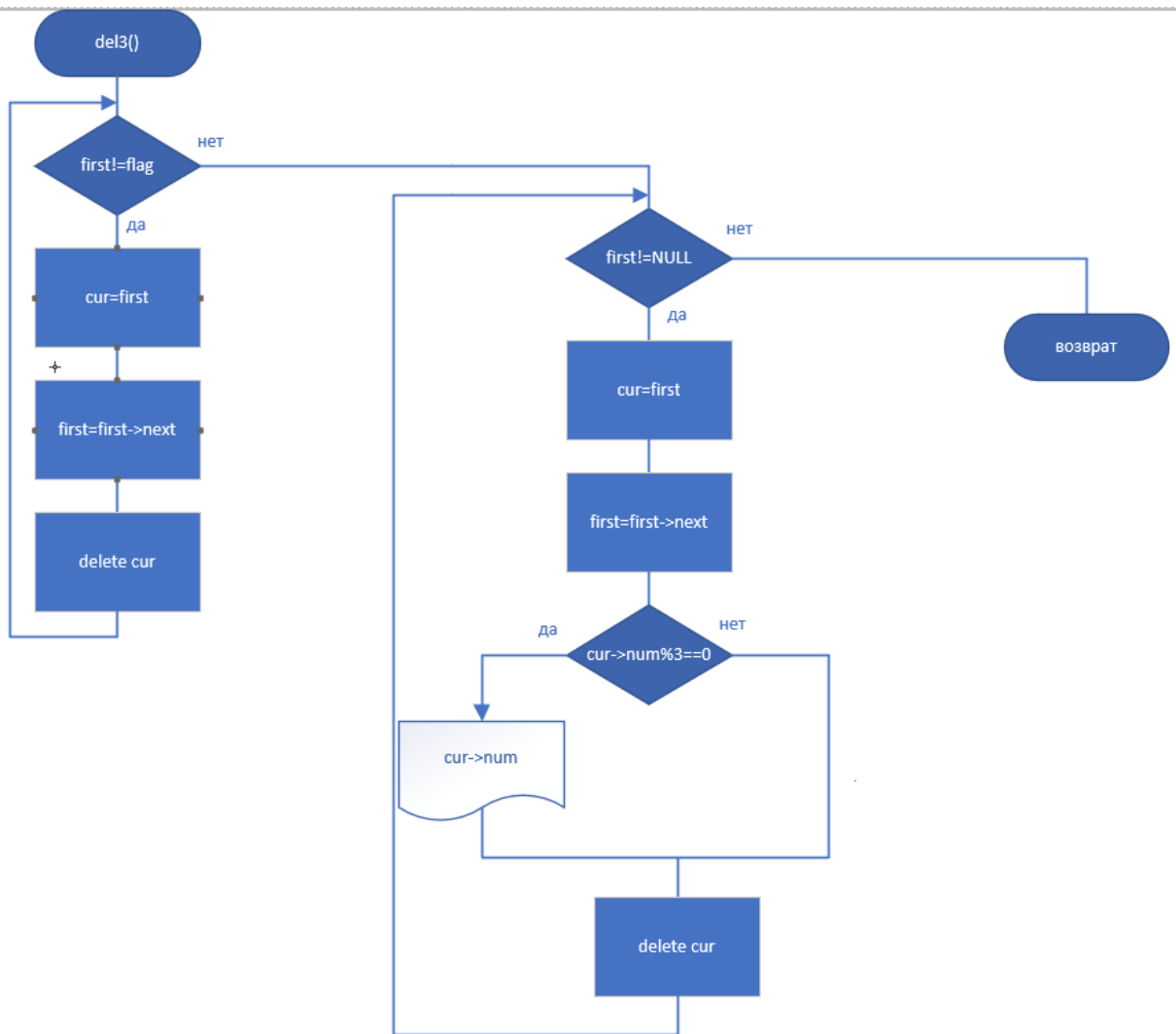
Цель: научиться работать с динамическими структурами данных, в частности, со списками.

Задание: Сформировать список из последовательности натуральных чисел $r_1, r_2, r_3, \dots, r_n$, которые вводятся с клавиатуры. Вывести данные числа сначала в обратном порядке, а затем в порядке ввода. Из полученного списка удалить все числа, кратные 3 и вывести полученный список.

Схема алгоритма







Текст программы

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
struct list {
    int num;
    list *next;
};
list *first,*last,*flag;
void create() {
    char str[7];
    gets_s(str);
    first=new list;
    first->num=atoi(str);
    last=first; flag=first;
    while (gets_s(str),strlen(str)!=0) {
        list *cur;
        cur=new list;
        cur->num=atoi(str);
        cur->next=NULL;
        last->next=cur;
        last=cur;
        cur=new list;
    }
}

```

```

        cur->num=atoi(str);
        cur->next=first;
        first=cur;
    }
}
void del3() {
    while (first!=flag) {
        list *cur=first;
        first=first->next;
        delete cur;
    }
    while (first!=NULL) {
        list *cur=first;
        first=first->next;
        if (cur->num%3!=0) printf("%7d",cur->num);
        delete cur;
    }
}
void main() {
    puts("Input numbers");
    create();
    list *cur=first;
    puts("Inverse order:");
    while (cur!=flag->next) {
        printf("%7d",cur->num);
        cur=cur->next;
    }
    cur=flag;
    printf("\n");
    puts("Standard order:");
    while(cur!=NULL) {
        printf("%7d",cur->num);
        cur=cur->next;
    }
    printf("\n");
    puts("Multiple 3 numbers:");
    cur=flag;
    while(cur!=NULL) {
        if (cur->num%3==0) printf("%7d",cur->num);
        cur=cur->next;
    }
    printf("\n");
    puts("Without multiple 3:");
    del3();
    printf("\n");
    system("pause");
}

```

Тестирование

```
C:\Users\Knigan\source\repos\Project1\Debug\Project1.exe
Input numbers
-8
-7
-6
-5
-4
-3
-2
-1
0
1
2
3
4
5
6
Inverse order:
  6    5    4    3    2    1    0   -1   -2   -3   -4   -5   -6   -7   -8
Standard order:
-8   -7   -6   -5   -4   -3   -2   -1    0    1    2    3    4    5    6
Multiple 3 numbers:
-6   -3    0    3    6
Without multiple 3:
-8   -7   -5   -4   -2   -1    1    2    4    5
Для продолжения нажмите любую клавишу . . .
```

```
C:\Users\Knigan\source\repos\Project1\Debug\Project1.exe
Input numbers
21
13
9
23
29
32
Inverse order:
 32   29   23    9   13   21
Standard order:
 21   13    9   23   29   32
Multiple 3 numbers:
 21    9
Without multiple 3:
 13   23   29   32
Для продолжения нажмите любую клавишу . . .
```

```
C:\Users\Knigan\source\repos\Project1\Debug\Project1.exe
Input numbers
13
64
37
15
2
-59
Inverse order:
-59 2 15 37 64 13
Standard order:
13 64 37 15 2 -59
Multiple 3 numbers:
15
Without miltiple 3:
13 64 37 2 -59
Для продолжения нажмите любую клавишу . . .
```

```
C:\Users\Knigan\source\repos\Project1\Debug\Project1.exe
Input numbers
1
2
4
5
3
6
-1
-6
-2
-3
-4
Inverse order:
-4 -3 -2 -6 -1 6 3 5 4 2 1
Standard order:
1 2 4 5 3 6 -1 -6 -2 -3 -4
Multiple 3 numbers:
3 6 -6 -3
Without miltiple 3:
1 2 4 5 -1 -2 -4
Для продолжения нажмите любую клавишу . . .
```

```
C:\Users\Knigan\source\repos\Project1\Debug\Project1.exe
Input numbers
5
7
9
15
27
13
46
64
0
-8
-7
-2
1

Inverse order:
1 -2 -7 -8 0 64 46 13 27 15 9 7 5
Standard order:
5 7 9 15 27 13 46 64 0 -8 -7 -2 1
Multiple 3 numbers:
9 15 27 0
Without multiple 3:
5 7 13 46 64 -8 -7 -2 1
Для продолжения нажмите любую клавишу . . .
```

```
C:\Users\Knigan\source\repos\Project1\Debug\Project1.exe
Input numbers
1
2
4
5

Inverse order:
5 4 2 1
Standard order:
1 2 4 5
Multiple 3 numbers:

Without multiple 3:
1 2 4 5
Для продолжения нажмите любую клавишу . . .
```

```
C:\Users\Knigan\source\repos\Project1\Debug\Project1.exe
Input numbers
1

Inverse order:
1
Standard order:
1
Multiple 3 numbers:

Without multiple 3:
1
Для продолжения нажмите любую клавишу . . .
```



```
C:\Users\Knigan\source\repos\Project1\Debug\Project1.exe
Input numbers
0
Inverse order:
0
Standard order:
0
Multiple 3 numbers:
0
Without multiple 3:
Для продолжения нажмите любую клавишу . . .
```

```
C:\Users\Knigan\source\repos\Project1\Debug\Project1.exe
Input numbers
List is empty
Для продолжения нажмите любую клавишу . . .
```

Вывод: я научился работать с динамическими структурами данных и, в частности, научился работать со списками.

Ответы на контрольные вопросы

1. Что такое «список» в программировании? В каких случаях используется эта конструкция? (Ответ: список – структура, при организации которой использованы указатели, содержащие адреса следующих элементов)
2. Какие типы списковых структур вы знаете? (Ответ: очередь, стек, дек)
3. Что такое указатель и как они объявляются? (Ответ: указатель – это переменная, в которой записан адрес ячейки памяти)
4. Как описывается элемент списка (Ответ: с помощью указателей списка)
5. Какие варианты возможны для добавления элемента к списку? (Ответ: добавить элемент к началу или к концу списка)
6. Как отлаживают программы, содержащие обработку списков? (Ответ: используют такие входные данные, чтобы проверить обработку списка во всех возможных вариантах выполнения программы)